College Library Management

Thesis · April 2012

CITATIONS	READS
0	150,107

3 authors, including:

Avijit Mathur
Intel India
25 PUBLICATIONS   **198** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Secure Solution for an end-to-end IoT system View project

INTRODUCTION

Project Overview

This year the summer training project titled 'College Library Management' is a vast one with the usage of numerous resources and popular software like the Microsoft Visual Studio 2008 and SQL server management to build a website for the easiness of the library staff.

The project's main aim is to build a management system more like a website which

- Can create a computerized management system for a library
- Has the capability to issue books
- Has the capability to return books
- Has an administrator account
- Can manage users through administrator account
- Has customized profile with photo of the user
- Tracks the books that users have issued
- Keeps the databases correct and up-to-date
- Can store all the book and user data in a proper manner

With the blooming of technology I thought it would be of a good sense to automate some of our oldest systems. So I asked myself which system is

- Not up-to-date
- Requires management
- Has its components value very high in the education department

That's when I came up with a plan to create a website 'College Library Management' by doing an internship in Rhydo technologies, where I learned

- Visual C#
- Using Microsoft Visual Studio
- Using SQL server management

- Embedding graphic CSS templates in my website
- Various dynamics of creating web pages and connecting them in order to get a website

Before going into any details I would like to explain exactly what this website does and why is it important

The website consists of several web pages having different

- Views
- Functionality
- Purpose
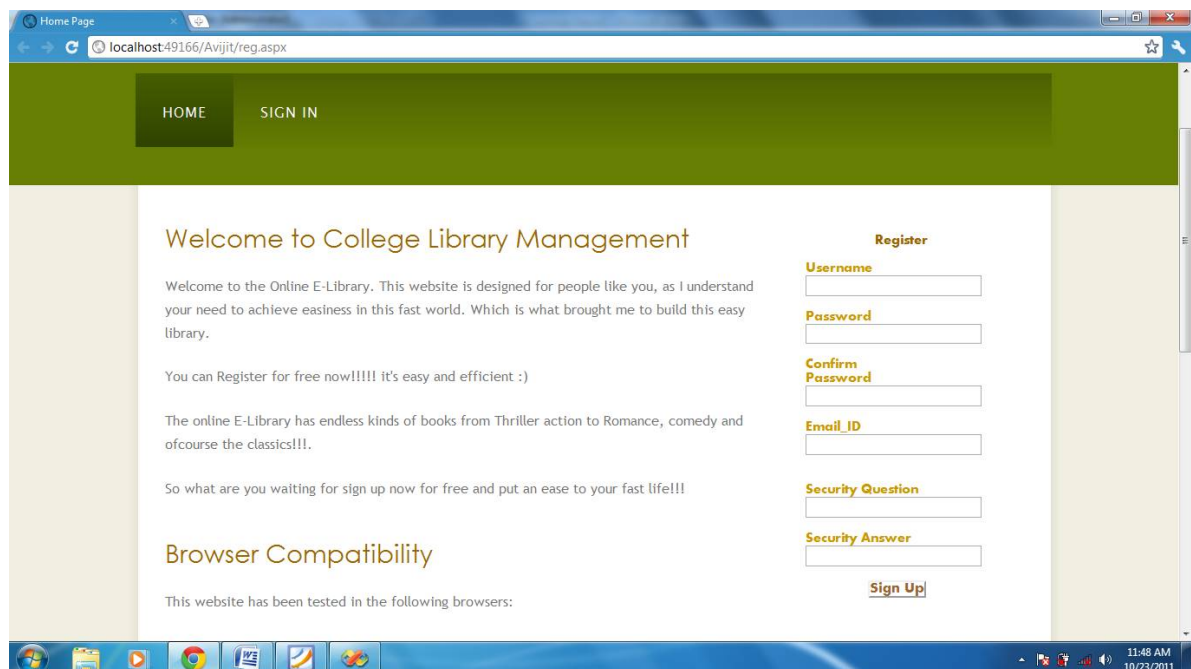
The following web pages constitute the 'College Library Management' website

- Main Page
    - Consisting of the WELCOME and REGISTRAION columns
- Login Page
    - Includes an interface to Login both for users and administrator
    - A forgot password facility
- Admin Page
    - Includes links to other pages: 'Add Books', 'Remove Books', 'Manage Users'

- - Has a welcome page with profile options
  - A Log out tab
- User Page
  - Includes links to other pages: 'Issue Books', 'Account', 'Return Books'
  - And a welcome page with profile options
  - A Log out tab
  -

All these pages are interlinked using Microsoft visual studio 2008 and the coding which it uses is that of C#. The data is held in databases managed by the SQL server management 2005 which gives the facility to connect the database to the website so that the data can be viewed, updated or removed as required.

Product Functions

The College Library System provides College real time information about the books available in the Library and the user information. The functions of the system include the system providing different type of services based on the type of users [Member/Admin].

- The member should be provided with the updated information about the books catalog.
- Provisions for the members to borrow the books they want, if all the other required rules hold good.
- The member is given a provision to check his account information and change the account information any time in the given valid period.
- The members are provided with the books available roster and allowed to choose the books, which they want to use in the coming up days.
- The admin can get the information about the members who have borrowed or returned the books.

- The admin is provided with interfaces to add/delete the books available in the book catalog.

User characteristics

The users of the system are members and the administrator who maintains the system. The member is assumed to have basic knowledge of the computers and Internet browsing. The administrator of the system has more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user's manual, College help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

Constraints

- The information of all the users must be stored in a database that is accessible by the College Library System. .
- The users must have their correct usernames and passwords to enter into the College Library System.

Assumptions and dependencies

- The users have sufficient knowledge of computers.
- The users know the English language, as the user interface will be provided in English
- The product can access the student database

Document Conventions

The following are the list of conventions and acronyms used in this document and the project as well:

Admin: A login id representing a user with user administration privileges to the software

User: A general login id assigned to most users

Client: Intended users for the software

SQL: Structured Query Language; used to retrieve information from a database

SQL Server: A server used to store data in an organized format

ASP: Active Server Pages: A Web Page formatted on the server and delivered to the browser.

Layer: Represents a section of the project

User Interface Layer: The section of the assignment referring to what the user interacts with directly.

Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed.

Data Storage Layer: The section of the assignment referring to where all data is recorded.

Data flow diagram: It shows the dataflow between the entities.

Use Case: A broad level diagram of the project showing a basic overview

Boolean: A true/false notation

Interface: Something used to communicate across different mediums

Unique Key: Used to differentiate entries in a database

FEASIBILITY STUDY

In feasibility study phase various steps have been taken:

1. Identify information at different levels.
2. Identify the expectation of user from an automated system.
3. Analyze the importance of automated system

Feasibility study:

In order to make sure that the project is feasible, following feasibility studies have been conducted: -

Technical Feasibility study:

It is possible to develop the system using simple platform. All the functions of a project for communication can be implemented in the new system. Hence the system is technically feasible.

Economic Feasibility study:

College library management system is a worth making project. This project is economically feasible in every sense that it takes less effort, less time, and nominal cost of purchasing the tools for developing the software.

Legal feasibility study:

Since the project needs no copyright, patenting, and doesn't intent to have any relation with anybody else's intellectual property rights, it can be considered as a legally feasible project.

Time feasibility study:

As it has been more probable (as per the requirements, functions, and performance specifications of the system) that the project can be completed within the given time

frame, it is considered that the undertaking this project is feasible in the context of time.

Operational feasibility study:

This system is completely operational and can be successfully implemented. College library management system is easy to understand not only for any sophisticated users but for the naïve users as well .It provides simple ambience in which user can feel free to work faster, easier, and more accurately. Therefore, it can be socially and behaviorally accepted and is feasible too.

Project planning:

During Planning all the activities that are to be performed to create the system are planned. Following are some of the issues that are well devised so that proper monitoring and controlling of the project could be easily done: -

Project development:

To avoid being stuck in dilemma during the development of project, one need to be sure that the process model he's using is right for the project. Since all the requirements about the problem can't be easily understood and may not be stable during the development of the system.

Quality Planning:

To ensure that the final software for-"College library management system" is of high quality, some quality control activities should be decided /planned in advance to perform them throughout the development of the software.

Following is a list of quality control activities that are used to identify and remove defects from the software, hence making it a high quality controlled system: -

Requirements Review

Design Reviews

Code Reviews

Testing

Risk control planning:

A risk is a probabilistic event – it may or may not occur. The aim of risk control planning is to minimize the impact of risks (if they occur) in the project.

Following are some known risks that might occur and their mitigation plan: -

Unclear project requirements ~> Keep in touch with the faculty in charge.

Data loss ~> Use CD's and/or pen drives to have some extra backups of the data.

Project delays ~> Use proper scheduling of the project as soon as possible so that the project could be completed.

Software size planning:

It has been taken into account that there are some functions in project that are indispensable .And these should not be excluded from project .Such functions /modules are like login, Sign up, admin add, admin remove, add user, remove user. So there are at least a minimum number of modules that have to be there in project.

Effort / cost Estimation:

Project scheduling:

During early stages of project planning, a kind of macroscopic schedule was already planned which gave a rough idea about activities that should be carried out for developing the project .In project scheduling , those sets of activities are refined into a detailed schedule.

OVERALL DESCRIPTION

Product Perspective:

College library management system is a product which does not intent to have any relation with any other product. It is a complete system in itself. It is an exclusive product which is to be concerned with the optimization.

Product Features: The project mainly use the concepts of .Net, simple tools of programming and SQL server for database storage

User Classes and Characteristics

There are number of functions that the system/product is supposed to perform which is as follows: -

- Issue books
- Remove books
- Admin login
- User login
- Add books
- Welcome page
- New user

The user of this product need not be computer expert. Even a naïve user can also operate the system. The user interfaces are to be made so simple that anybody can be comfortable in working with the system in just a few minutes. The basic things which are required in a user are:

- User should know what the computer is.
- User should understand English.
- User must know which key (button or keyboard) does what.
- User must have an experience of library management.

Operating Environment:

The product will be operating in windows environment. Also it will be compatible with the higher version of this explorer.

Design and Implementation Constraints:

The general constraints which are to be introduced in project are as follows: -

- Only that person can operate the system's who knows the ID and Password of the valid user of the system.

- In case admin does not remembers his/her ID/Password then system won't consider him/her as a valid user for College library management system.

.

PERFORMANCE REQUIREMENT

Static Requirements:

Project College library management system is to support many users at a time. All necessary operations shall have been carried out with the help of many clients and a server.

Dynamic Requirements:

Project College management system has to avoid degradation of its performance by processing one or at most two forms at a time. If the user finishes his work related to one form and opens the other, then the previous form will be unloaded in order to save some memory space.

Logical Database Requirements:

All the information should be stored in separate databases. These databases should be categorized and maintained in a logical manner. For example:-

- User information in User database.
  - Administrator information in Admin database.

Design Constraints

.NET will have to be operated on Windows XP, VISTA, or 7. It will require at least 512MB of RAM, and few Megabytes of Hard Disk. As the security feature included in the system, no one except the administrator and member can access the system.

SOFTWARE SYSTEM ATTRIBUTES

Following are some of the basic attributes that the project should have:-

Reliability: - The system should be reliable in the sense that there should be no room for mistakes. Every activity/ function of project should be indefectible.

Availability: - The system is to be available as and when needed.

Security: - The system is to be secure in the sense that nobody except the authenticated users can login and use the system.

Portability: - The project should be made such that there are as less operating system dependencies as possible, or the software should be portable with only few modifications.

Maintainability: - The system should be maintainable in the sense that if any error occurs, it should be easily rectified and the cost incurred in maintenance should be as low as possible.

Organizing Specific Requirements

There are some requirements of the system that are indispensable while the others are lesser. Following is the organization of specific requirements in descending order of their importance: -

i.) Functional requirements.   Higher
ii.) External Interface requirements.
iii.) Performance requirements.
iv.) Design Constraints.
v.) Logical Database requirements.
vi.) Software system attributes.   Lower

Description

The user interface must be customizable by the administrator

Criticality

This issue is essential to the overall system. All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined.

Technical issues

In order to satisfy this requirement the design should be simple and all the different interfaces should follow a standard template. There will be the possibility of changing colors and images, plus switching between interfaces with the minimum impact for the users.

Risks

To reduce the circumstances under which this requirement might not able to be satisfied, all the designers must have been developed web sites previously and they must be aware of html restriction and cross browsers implementations before starting the designing. In order to reduce the probability of this occurrence the entire design team will be trained in basic html development and macromedia fireworks, this tool will be used instead of Photoshop.

Dependencies with other requirements

All user interfaces should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module.

METHODOLOGY ADOPATED

To make sure that the project 'College library management system faces no probabilities of being disorganized, unsystematic, and/ or having undesirable consequences, a clear perspective has been defined by consenting to use a model for this project. The model that is chosen is called 'Spiral Model'.

According to a Software Engineering book, written by Roger S. Pressman, the Spiral Model has five phases: -

1. Communication Phase
2. Planning Phase
3. Modeling Phase
4. Construction Phase
5. Deployment Phase

The project has been following this methodology because all the requirements about the problem can't be easily understood and may not be stable.

Representation of Spiral Model
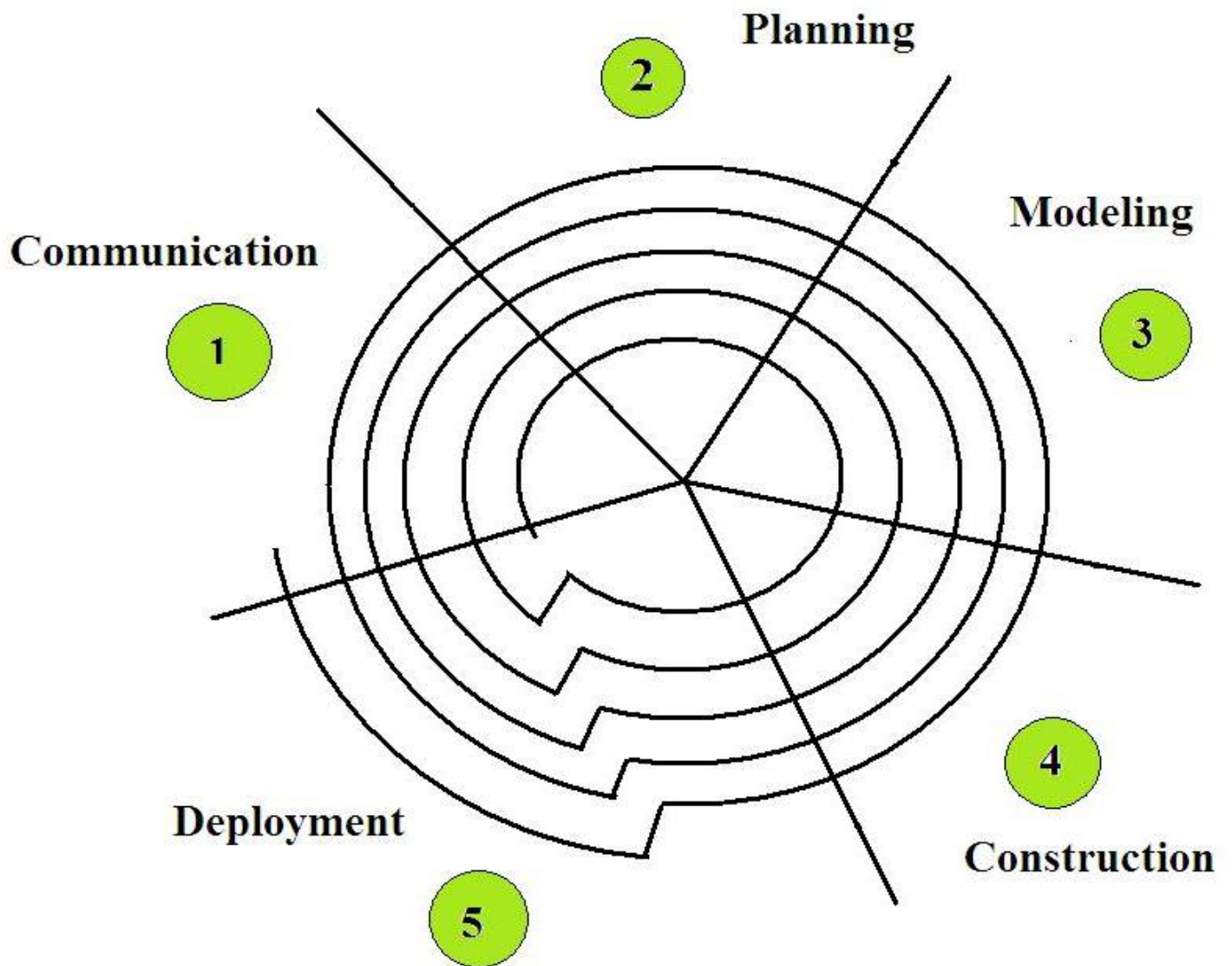
Communication Phase: -

Before the requirements can be analyzed, modeled, or specified they must be gathered through a communication activity.

Under this phase, all kinds of information have been gathered to successfully make the project of the desired specifications.

It includes: -

- On-site Observations.
- Informal Meetings
- Personal discussions
- Reviews of Information Domains
    - Admin Report
    - User Report

Planning Phase: -

The planning activity encompasses a set of management and technical practices that enable the software team to define a road map as it travels toward its strategic goal and tactical objectives. It includes the issues like: -

- Clarification of the reasons for developing the system.
    - To automate communication between various computers.
    - To gain better experience in IT field

- Identification of the functionalities to be built.
    - Security Features.
    - Easy and Fast Procedures for Doing Typical Operations on any local area network.

- Proper scheduling of the system.
    - One Week (Approx.) For Completing the Synopsis.
    - One Month (Approx.) For Developing the System.

Modeling Phase: -

System modeling is an important element of the system engineering process. Under this phase, models are developed to gain a better understanding of the actual entity to be built.

Construction Phase: -

The construction activity encompasses a set of coding and testing tasks that lead to operational software that is ready for delivery to the customer or end-user.

- Coding
  - The Creation of Programs/Database Using codes and Statements in C# and SQL.

- Testing
  - After The Programs Get Created, They Are Tested With The Intent Of Finding Any Error That Was Left Undetected During the Coding Phase. The Basic Tests That Have Been Done Are: -

    - Unit Testing
    - Integration Testing
    - System testing
    - Alpha Testing
    - Beta Testing
    - Acceptance Testing

Deployment Phase: -

This phase encompasses following activities: -

- Delivery of Software, Support For User and Feedback

HARDWARE DETAILS

<u>Processor</u>: **Intel Core 2 Duo T6600 @ 2.2GHz (2CPUs)**

Core 2 is a brand encompassing a range of Intel's consumer 64-bitx86-64 single-, dual-, and quad-core microprocessors based on the Core micro architecture. The single- and dual-core models are single-die, whereas the quad-core models comprise two dies, each containing two cores, packaged in a multi-chip module. The introduction of Core 2 relegated the Pentium brand to the mid-range market, and reunified laptop and desktop CPU lines, which previously had been divided into the Pentium 4, Pentium D, and Pentium M brands.

<u>Hard Disk</u>: **500 GB**

A hard disk is part of a unit, often called a "disk drive", "hard drive", or "hard disk drive", that stores and provides relatively quick access to large amounts of data on an electromagnetically charged surface or a set of surfaces. A hard disk is really a set of stacked "disks", each of which, like phonograph records, has data recorded electro-magnetically in concentric circles or "tracks" on the disk. A "head" writes or reads the information on the tracks. Two heads, one on each side of a disk, read or write the data as the disk spins. A hard disk/drive unit comes with a set rotation speed varying from 4500 to 7200 the physical location can be identified with the cylinder, track, and sector locations, these are actually mapped to a Logical Block Address (LBA) that works with the larger address range on today's hard disks.

<u>Random Access Memory</u>: **3 GB**

When people talk about computer memory, they usually mean the volatile RAM memory. Physically, this memory consists of some integrated circuit

Chips (IC chips) either on the motherboard or on a small circuit board attached to the motherboard. A computer's motherboard is designed in a manner that its memory capacity can be enhanced by adding more memory chips. Hence, if you decide to have

more memory than your computer currently has, you can buy more memory chips, and plug them in the empty memory slots on the motherboard. This job is normally done by the service engineers.

Monitor: **LCD**

A **liquid crystal display** (**LCD**) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly.

It has a resolution of 1366 x 768 x 59 Hz

Keyboard: **Multimedia**

The multimedia keyboard is a kind of keyboard that gives you a superior level of desktop freedom to work anywhere, on almost any surface. The Multimedia Keyboard provides complete workspace freedom. Easily navigate to the web, check your email, control your audio or access documents with the touch of a button on the multimedia keyboard. It features time saving multimedia and internet shortcut buttons including back, forward, stop, refresh, web and search buttons for the internet as well as play controls, volume controls on Windows operating systems. The multimedia keyboard is compatible with almost all PC models

Mouse: **Optical**

An optical mouse is an advanced computer-pointing device that uses a Light Emitting Diode (LED), an optical sensor, and Digital Signal Processing (DSP) in place of the traditional mouse ball and electromechanical transducer. Movement is detected by sensing changes in reflected light rather than by interpreting the motion of a rolling sphere. The optical mouse takes microscopic snapshots of the working surface at the rate of more than 1,000 images per second. If the mouse is moved then the image changes. The tiniest irregularities in the surface can produce images well enough for the sensor and DSP to generate usable movement data.

SOFTWARE DETAILS

Operating System: **Windows 7**

Microsoft Corporation	
Releases	
Release date	**RTM version:** July 22, 2009 **Retail version:** October 22, 2009
Current version	6.1 (Build 7601: Service Pack 1) (February 22, 2011; 12 months ago)
Source model	Closed source / Shared source
License	Proprietary commercial software
Kernel type	Hybrid
Update method	Windows Update
Platform support	IA-32 and x86-64
Preceded by	Windows Vista
Succeeded by	Windows 8

<u>Front End</u>: - ASP.NET **2.0 with C#**

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic websites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages(ASP) technology.ASP.NET is build on the Common Language Runtime(CLR),allowing programmers to write ASP.NET code using any supported .NET Language.

C# is a **strongly-typed object oriented language** designed to give the optimum blend to simplicity, expressiveness, and performance. The .NET platform is centered around a Common Language Runtime(similar to a **JVM**) and a set of libraries which can be exploited by a wide variety of languages which are able to work together by all compiling to an intermediate language(IL). C# and .NET are a little symbiotic: some features of C# are those to work well with .NET, and some features of .NET are there to work well with C#(though .NET aims to work well with many languages).

<u>Back End</u>: **- Microsoft SQL Server 2005**

Microsoft SQL Server is a relational model database server produced by MICROSOFT. Its primary query languages are T-SQL and ANSI SQL.SQL Server allows multiple clients to use the same database concurrently.

As such, it needs to control concurrent access to shared data, to ensure integrity- when multiple clients update the same data, or clients attempt to read that is in the process of being changed by another client.

<h1 style="text-align:center">COST & BENEFITS ANALYSIS</h1>

COST

Various costs that are incurred in making the system are:-

1. Hardware Cost

    a. Computer purchase

    b. SQL server software purchase & installation.

    c. .NET software purchase & installation.

2. Facility Cost

    a. Proper lightning.

    b. Air Conditioning

3. Operating Cost

    a. Consumption of electricity

4. Supply Cost

    a. Compact Discs/Pen drives.

BENEFITS

Various benefits that are to be obtained with project are: -

1. Improved performance
   a. Accuracy improvement.
   b. Time Saving.
2. Decreased Supply cost
   a. No use of registers.
3. Decreased personnel cost
   a. Fewer staff.
   b. Fewer payments.

CONCLUSION OF THE COST AND BENEFIT ANALYSIS

Most of the costs that have been incurred in developing the system are onetime costs. While all the benefits that we would get are not for one time only, they shall be obtained on a regular basis. In this way, benefits will exceed costs by a substantial margin, hence this project can be considered as cost effective.

DETIALED LIFECYCLE OF PROJECT

The term "PROJECT LIFE CYCLE" refers to a sequence of pre-planned stages for taking a project from beginning to the end.

Just like any other project, this project also has a lifecycle which is broadly categorized into following five stages, viz.:

Conceptualization

Determining the need for developing Website for Multiplex theatres and conducting Technical, Economical, Legal, Operational, and social/Behavioral Feasibility Studies for the project.

Definition /Planning

Defining the operations, Performance Parameters, And Planning for costs, schedules, and programming tools.

Design

Designing the data structure, database architecture and User Interface for the software.

Implementation

Implementing the system at the user end and getting feedback for the same. making subtle Modifications In The System(If necessary).

Conversion

Modifications at the higher level will be done in future if is required so. (E.g. Modifications could be to introduce new modules into the system).

PROCESS INVOLVED

The various processes involved in making the software successfully running and documenting the project are as follows: -

- **Preliminary Investigation**

  - **Making enquiries about the current system's working.**
  - **Conducting feasibility studies for the project to be undertaken.**
  - **Deciding whether to undertake the project or not.**

- **System Requirement**
  - **Gathering the information about the need to build the project**

- **System Design**
  - **Designing Data Flow Diagrams (DFD's), data dictionary, databases schemas, user interfaces.**

- **System Development**
  - **Developing Databases, Creating Forms, Coding of software, and integrating the components.**

- **System Testing**
  - **Testing Individual Units of the software as and when they are created.**
  - **Testing the integration of units of the software.**
  - **Testing the whole functioning of the software.**
  - **Testing the software at the developer's end.**
  - **Testing informally the software with general people.**
  - **Testing the software at the client's end.**

- **Implementation**
  - **Installing the software at the user's side. Direct conversion approach has been taken.**
  - **Getting feedback from the user.**

DRAWBACKS OF PRESENT SYSTEM

Some of the problems being faced in manual system are as follows:

1. Database for books not maintained in an effective manner
2. Tracking of users and books not an easy task
3. Very slow compared to an automated system
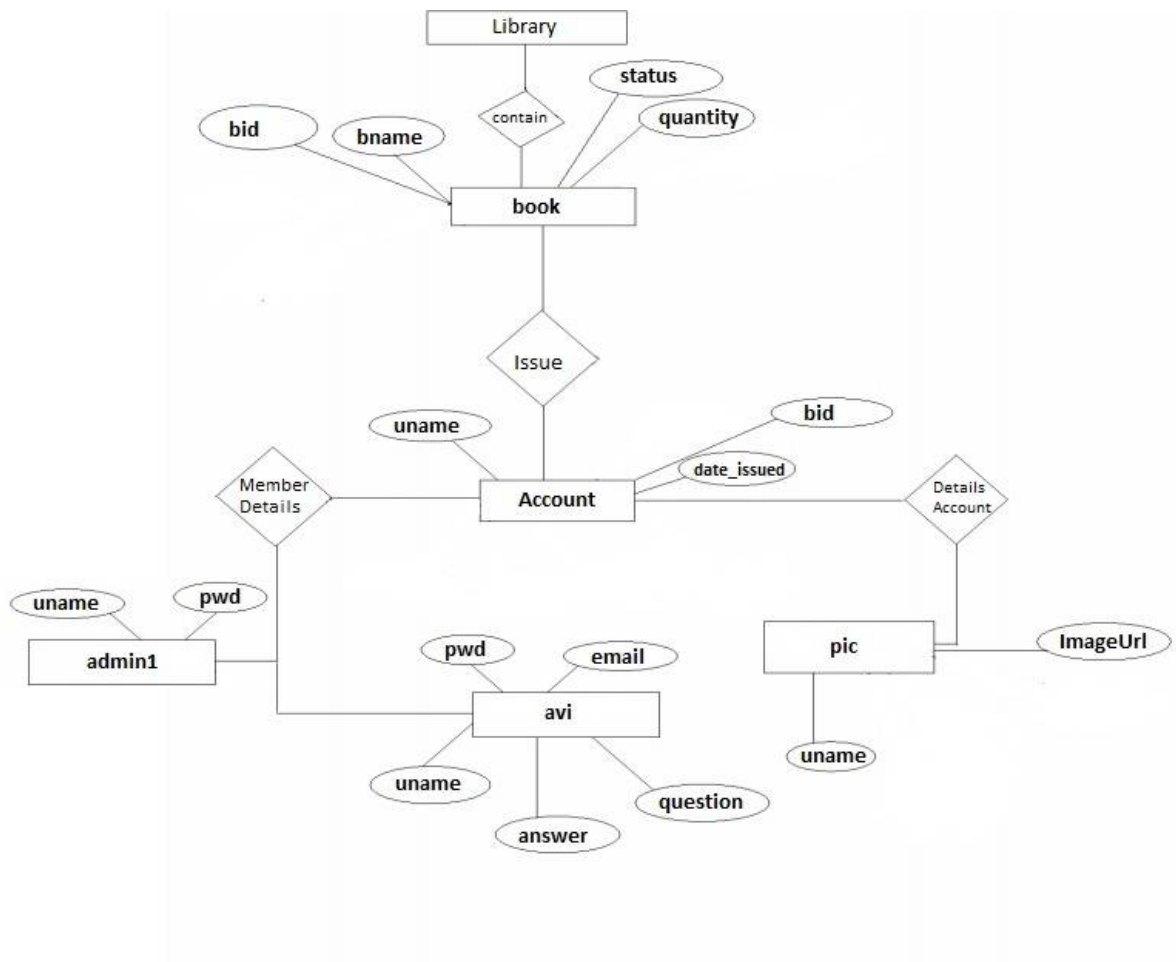4. Information about returning/issuing of books not maintained properly

Automated System

1. Books information maintained in a properly defined database
2. Ease of issuing and returning of books with a click of a button
3. Ease of tracking books issued by each member
4. Administrator can manage users and books effectively

DATA FLOW DIAGRAM
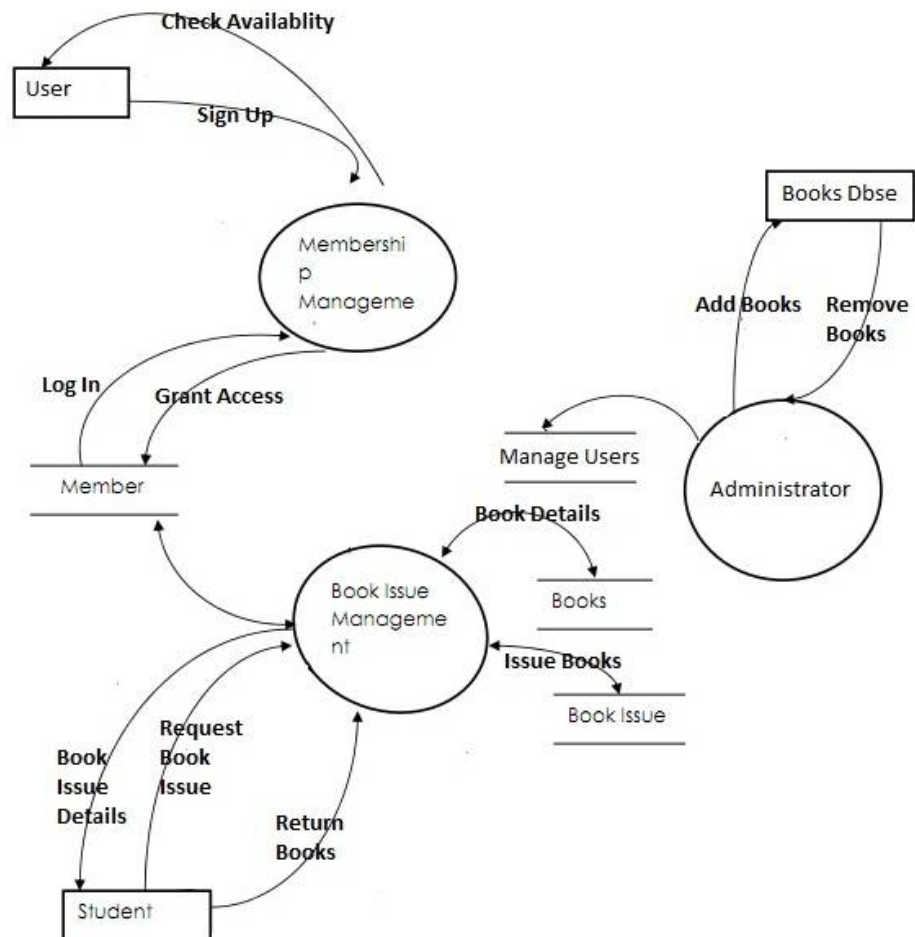
ER DIAGRAM

Purpose

The main objective of this document is to illustrate the requirements of the Library Management system. The document gives the detailed description of both functional and non functional requirements proposed. It describes the external behavior of the College Library System. Requirements Specification defines and describes the operations, interfaces, performance, and quality assurance requirements of the College Library System. The document also describes the nonfunctional requirements such as the user interfaces. It also describes the design constraints that are to be considered when the system is to be designed, and other factors necessary to provide a complete and comprehensive description of the requirements for the software. The Software Requirements Specification (**SRS**) captures the complete software requirements for the system, or a portion of the system.

Scope

The Software Requirements Specification captures all the requirements in a single document. The *College Library System* provides the members of the Library and employees of the library with books information, College managing of accounts and many other facilities. The College Library System is supposed to have the following features.

- The product provides the members with College adding and removal of books capabilities
- The system provides logon facility to the users.
- The system provides the members with the option to check their account and/or change their options like password and profile picture of the account whenever needed all through the day during the library hours.

- The system lets the administrator to check which all members have borrowed which all books

- The system allows the Librarian to create the books catalog, add/delete books and maintain the books catalog.

- The system updates the system as and when the member borrows or returns a book.

The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criterion for the system is based in the level up to which the features described in this document are implemented in the system.

Specific Requirements

This section describes in detail all the functional requirements.

Functionality

Logon Capabilities

The system shall provide the users with logon capabilities.

Catalog

The system shall provide with all the books available

Usability

- The system shall allow the users to access the system from a browser using HTML or it's derivative technologies.

- Since all users are familiar with the general usage of browsers, no specific training is required.

- The system is user friendly and self-explanatory.

Reliability

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Mean Time Between Failures (MTBF)

The system is developed in such a way that it **may** fail once in a year.

Mean Time to Repair (MTTR)

Even if the system fails, the system will be recovered back up within an hour or less.

Accuracy

The accuracy of the system is limited by the accuracy of the speed at which the employees of the library and users of the library use the system.

Maximum Bugs or Defect Rate

Not specified.

Access Reliability

The system shall provide 100% access reliability.

Performance

Response Time

The Information page should be able to be downloaded within a minute. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs.

Administrator/Librarian Response

The system shall take as less time as possible to provide service to the administrator or the librarian.

Throughput

The number of transactions is directly dependent on the number of users; the users may be the Librarian, employees of the Library and also the people who use the Library for checking-out books, returning books and checking College library account.
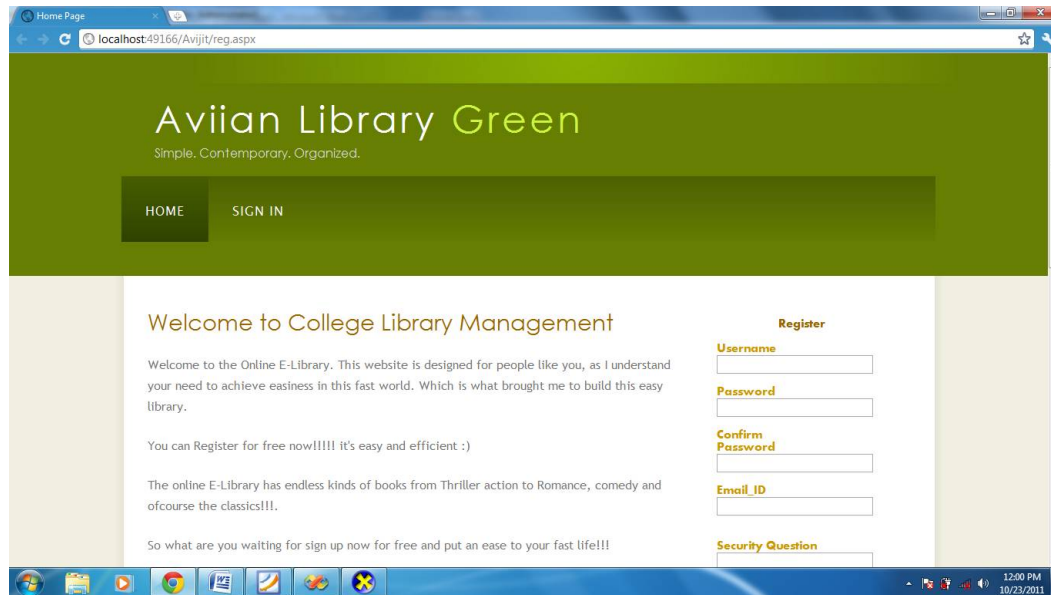
Capacity

The system is capable of handling many users.

Resource Utilization

The resources are modified according the user requirements and also according to the books requested by the users.
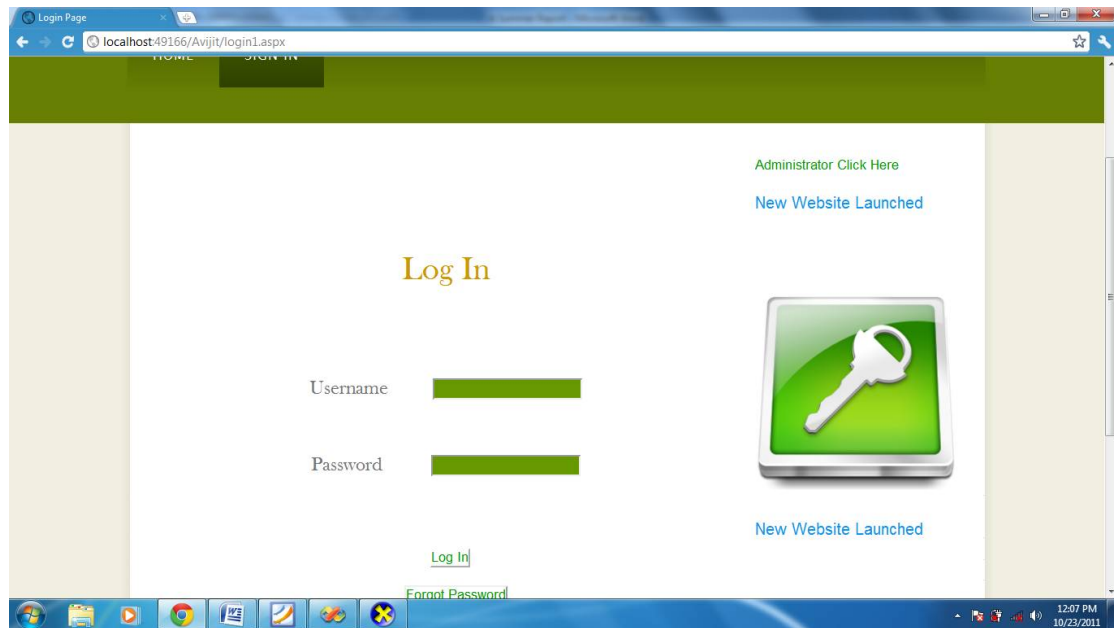
Home Page

The Home page consists of:

- The green background in which the name of my system 'Aviian Library Green' along with the tabs to the pages which blend in perfectly
- The welcome section introducing what the site is about
- The register column for easy signing up of a user within seconds
- A section showing browser compatibility
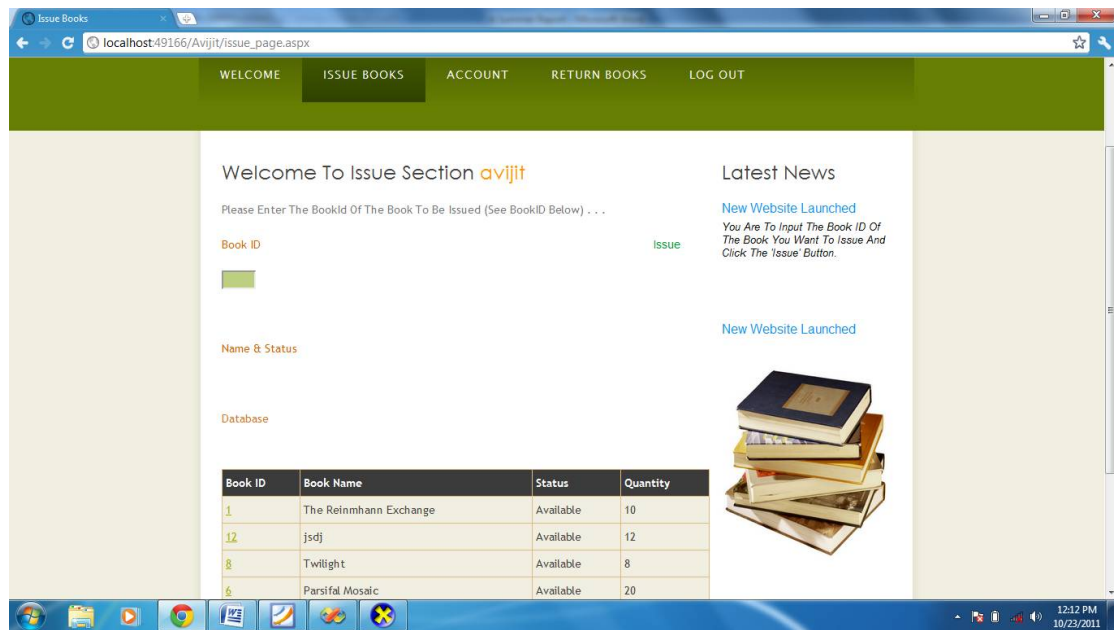
Login Page

The login page which includes:

- Login interface for users and administrators
- Easy link for administrators for logging in
- A forgot password facility which emails the password of the desired user

User Welcome Page
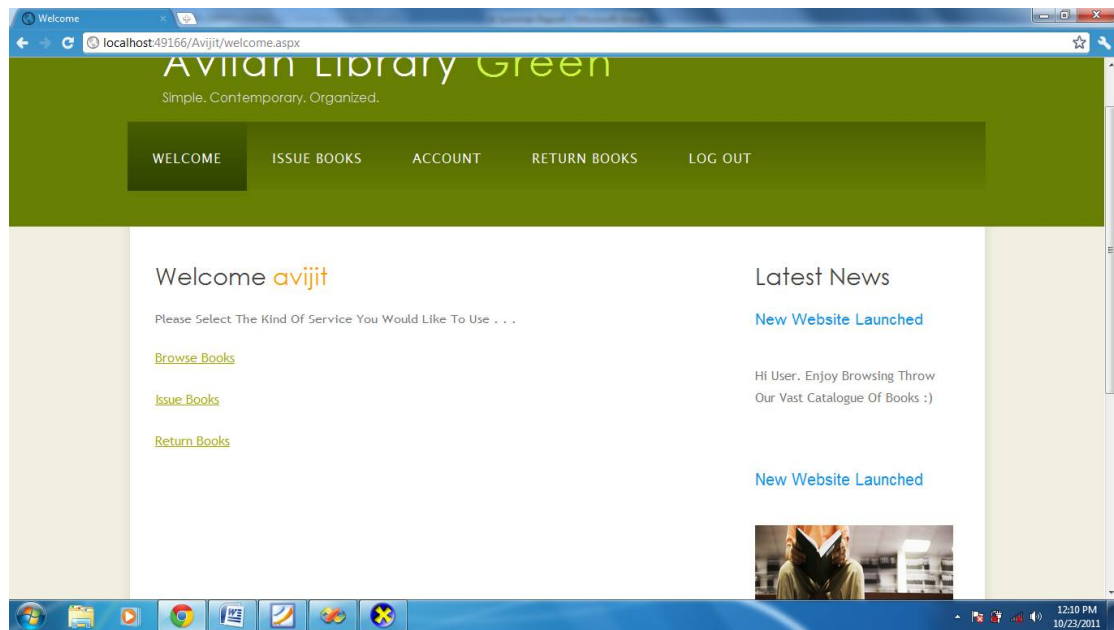
- Links and tabs for easy access of certain features for users
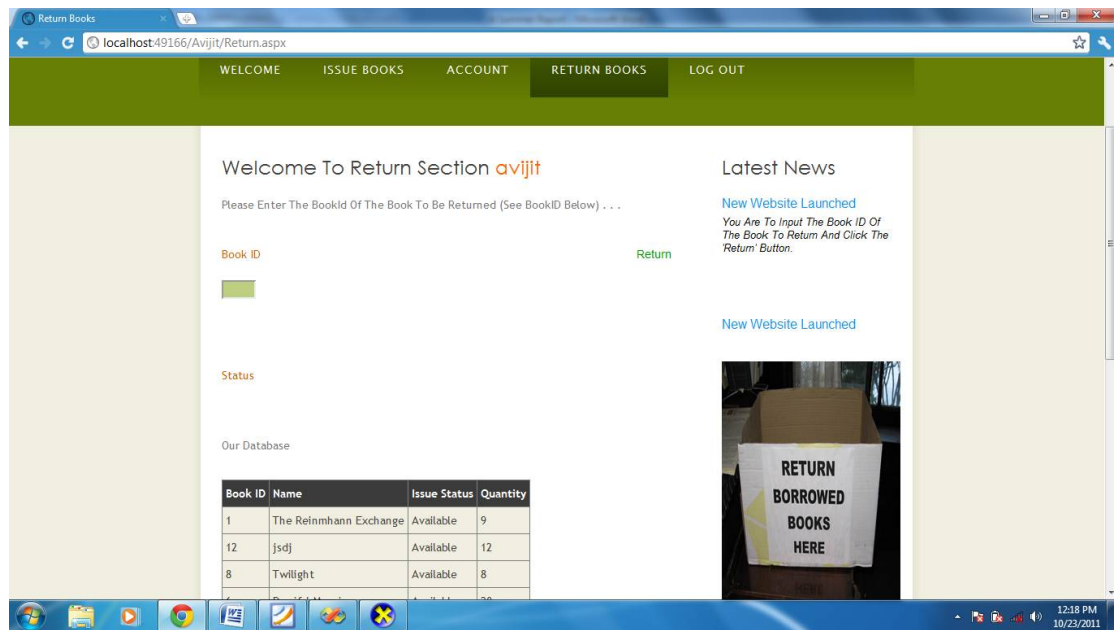- Name of the user in orange color

Issue Books Section

- Easy view of books available in the database
- Issue the books easily by just providing the Book ID of the book from the
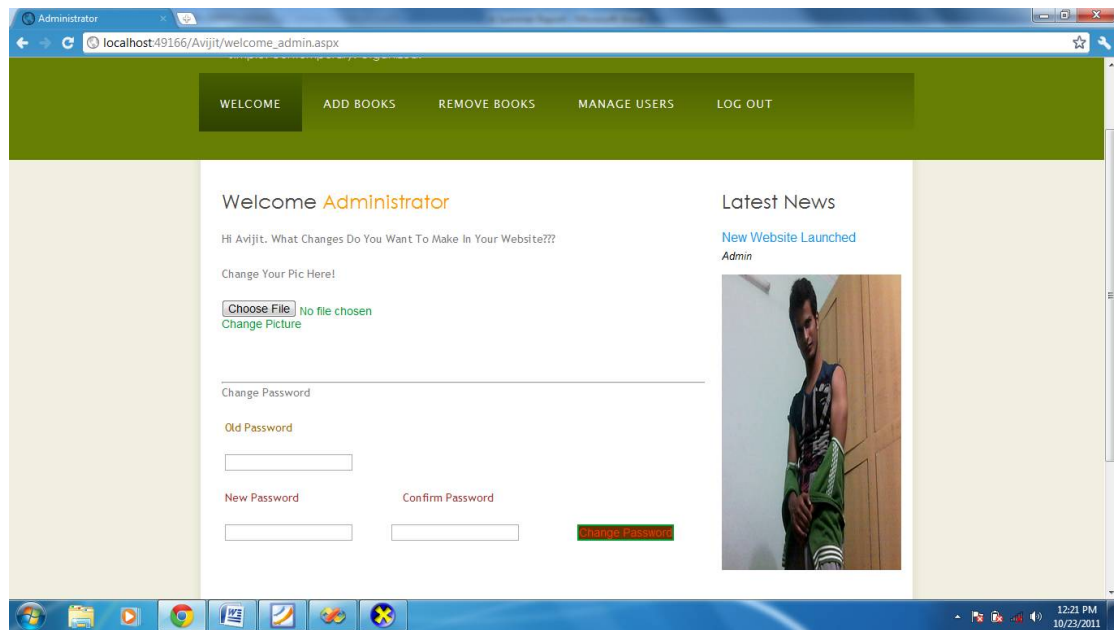 database

Account Section
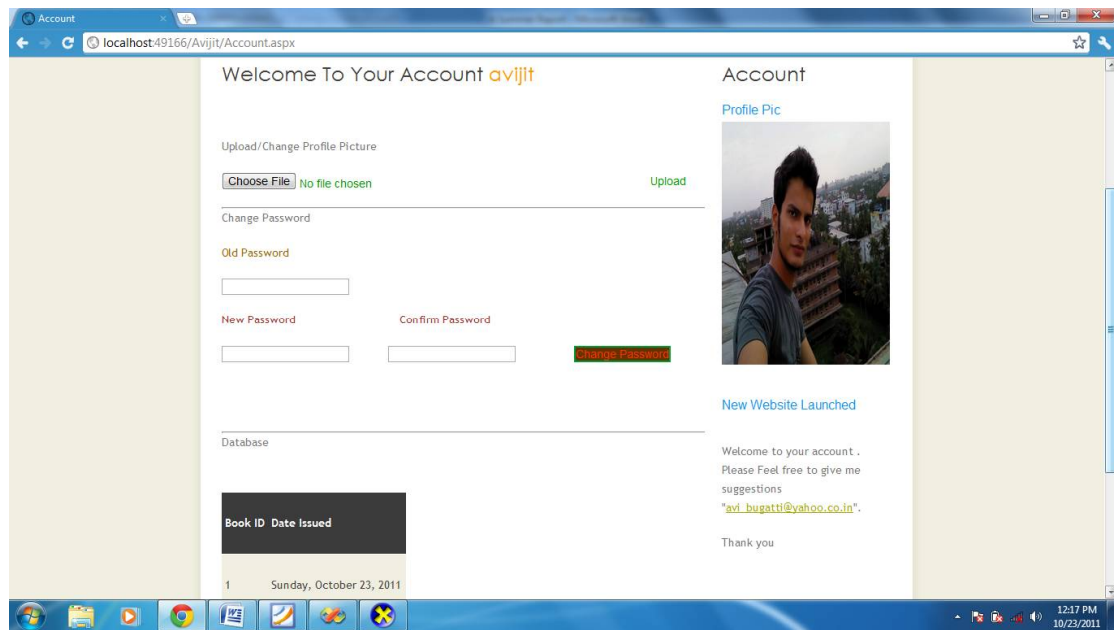
- Option to upload a new picture for the profile
- Password changing facility
- View of the books issued

Return Books Section

- Return the book by typing in the Book ID of the book to be returned
- View of database as it updates itself

Admin Welcome Page

- Change picture option
- Easy access to other pages through tabs
- Change password facility
- Picture of the admin

Add Books Section (admin)

- Feature to add a new book

- And extra copies of an existing book

- Database view of the copies of books

Remove Books Section
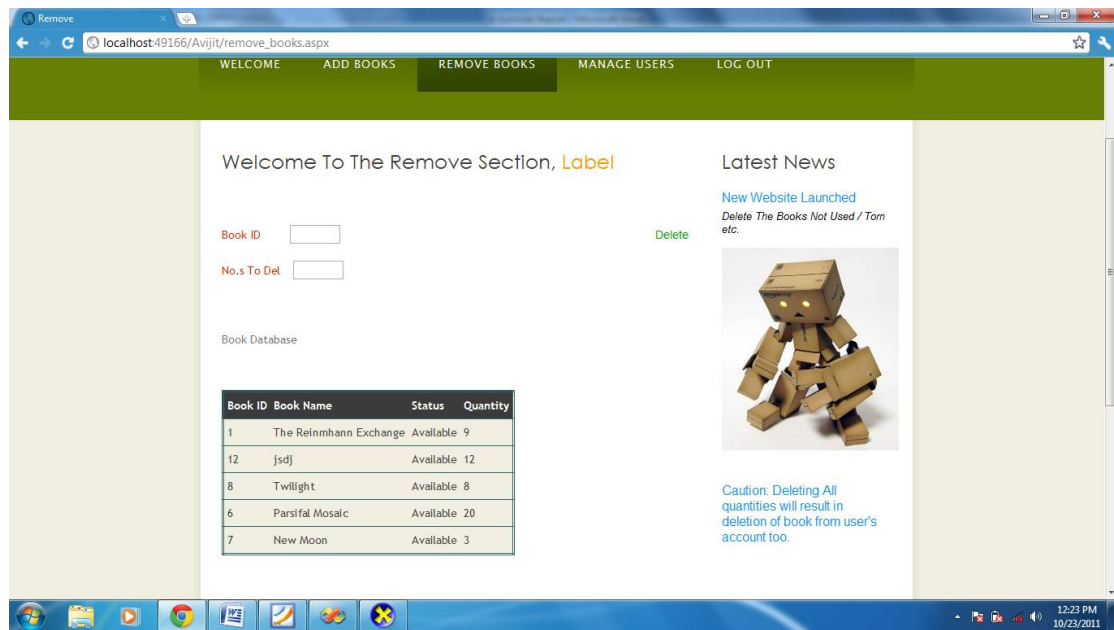
- Removal of books if required by the admin
- Database view

Manage Users (admin)
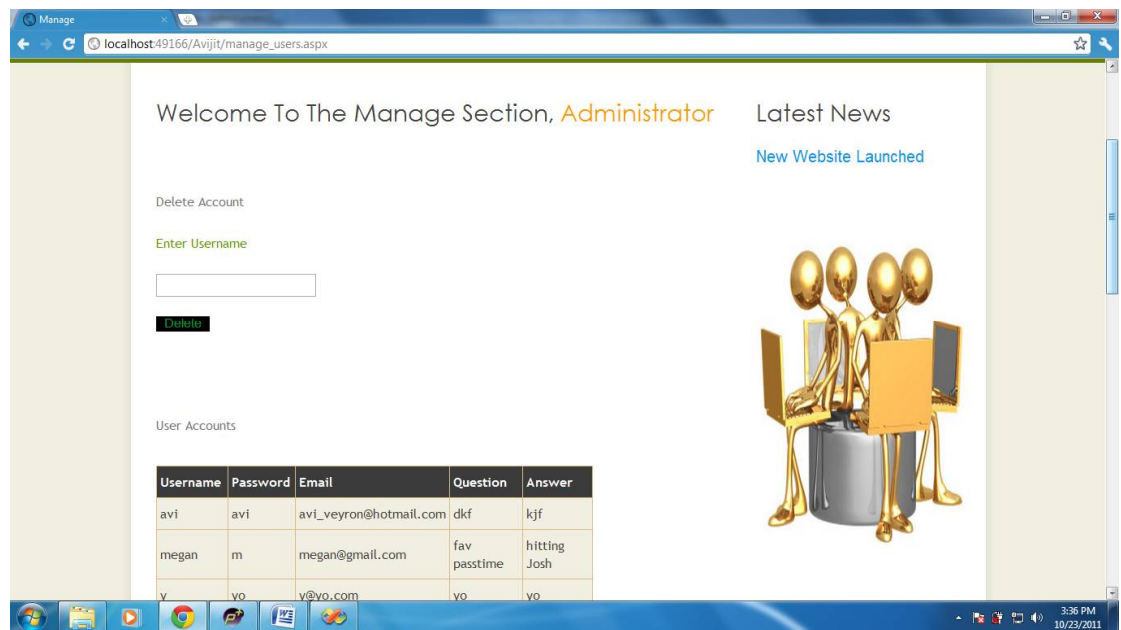

- Facility to delete a user (only admin has jurisdiction)
- Database view of user accounts, books and issue status

WORKING

CLASS

A class has been created for ease of use in the following pages to come.

```
/// <summary>
/// Summary description for Class1
/// </summary>
public class lib


{
        public lib()
        {


        }



    public SqlConnection con = new SqlConnection("Data Source= AVIJIT-
PC\\SQLEXPRESS;Initial Catalog=Avijit;Integrated Security=True");   //This can be
picked up from the connection string property of the connection to
//database created.
        public SqlCommand cmd = new SqlCommand();


    public SqlDataAdapter ad = new SqlDataAdapter();


    public DataTable dt = new DataTable();


    public SqlConnection getcon()
    {
        if (con.State == ConnectionState.Open)
        {
```

```csharp
                con.Close();
            }
        con.Open();
        return con;
    }



    public string excquery(string cmdtext)
    {
        try
        {
            cmd.Connection = getcon();


            cmd.CommandType = CommandType.Text;


            cmd.CommandText = cmdtext;


            cmd.ExecuteNonQuery();


            return "1";
        }


        catch (Exception ex)
        {
            return ex.Message;
        }
    }
    public string excproc(string procname)     // for stored procedures
    {
        try
        {
            cmd.Connection = getcon();
```

```csharp
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.CommandText = procname;

            cmd.ExecuteNonQuery();

            return "1";
        }

        catch (Exception ex)
        {
            return ex.Message;
        }
    }

    public DataTable gettable(string fetchcmd)
    {
        cmd.Connection = con;

        cmd.CommandType = CommandType.Text;

        cmd.CommandText = fetchcmd;

        ad.SelectCommand = cmd;

        dt = new DataTable();

        ad.Fill(dt);

        return dt;

    }
}
```

HOME PAGE WORKING

The home page is made using the dual concept of

> ➢ Master Page
> ➢ Content Page

The master page allows a consistent layout for the web page to be integrated into the website making it easier to safeguard the main layout without accidently editing it along with the contents that one needs to display.

A master page is an ASP.NET file with the extension .master (for example, library.master) with a predefined layout that can include static text, HTML elements, and server controls. The master page is identified by a special @Master directive that replaces the @Page directive that is used for ordinary .aspx pages. The directive looks like the following.

```
<%@ Master Language="C#" %>
```

The content page is used to display the content that the programmer wants the user to usually interact with or manipulate accordingly.

You define the content for the master page's placeholder controls by creating individual content pages, which are ASP.NET pages (.aspx files and, optionally, code-behind files) that are bound to a specific master page. The binding is established in the content page's @ Page directive by including a MasterPageFile attribute that points to the master page to be used. For example, a content page might have the following @ Page directive, which binds it to the library.master page.

```
<%@ Page Language="C#" MasterPageFile="~/MasterPages/library.master"
Title="Content Page"%>
```

Process:

1. Right click add new item to the website

2. Select master page along with the language 'Visual C#'

3. Add the template for graphic design of the web page

4. Then right click to add content page

Every page has a menu bar with the source code:

```
<li><a href="reg.aspx">Home</a></li>
<li class="selected"><a href="login1.aspx">Sign In</a></li>
```

In the content page text fields, labels and buttons were added in this fashion

FIG 1

➢ Color – Brown : Labels for the textfields

➢ Color – Reddish orange : Validators

Use of validators

I. <u>Required field validator</u> (the red *): Used to check whether a field is non-empty. The *ControlToValidate* property is set to the textfield to be checked.

II. <u>Compare Validator</u> (the 'passwords don't match'): Used to compare two pre specified fields and give a pre set result.The *ControlToValidate* property is used to set the first textfield and the *ControlToCompare* is used to set to the field to which the value is to be compared. *Type* property is used to set the type of the values to be compared (in this case: string).

III. <u>Regular Expression Validator</u>(the 'invalid email'): The *ControlToValidate* property is set to the textfield to be checked. The *ValidationExpression* property is set to the desired type of expression ('Internet e-mail address' in this case) and the *ErrorMessage* property to display the string.

```
public partial class _Default : System.Web.UI.Page
{
    lib obj = new lib();  // Uses the class created (lib)


    protected void signin_Click1(object sender, EventArgs e)
    {
        // Appending of the text written in the textfields of fig 1
        // As soon as the button with label sign up is clicked the data is checked, verified and written to the
        // Avijit database.

        obj.cmd.Parameters.AddWithValue("@uname", uname.Text);
        obj.cmd.Parameters.AddWithValue("@pwd", pwd.Text);
        obj.cmd.Parameters.AddWithValue("@email", email.Text);
        obj.cmd.Parameters.AddWithValue("@question", question.Text);
        obj.cmd.Parameters.AddWithValue("@answer", answer.Text);


        string s = obj.excproc("register");
```

```csharp
        if (s == "1")
        {
            Response.Redirect("login1.aspx");
        }
        else
        {
            Response.Write("Failure");
        }
    }


    protected void uname_TextChanged(object sender, EventArgs e)
    {

    }
}
```

The making of the Menu Bar

```html
<div id="menubar">
    <ul id="menu">
     <!-- class="selected" is put in the li tag for the selected page - to highlight
which page you're on -->
        <li class="selected"><a href="reg.aspx">Home</a></li>
        <li><a href="login1.aspx">Sign In</a></li>
      </ul>
     </div>
```

LOGIN PAGE WORKING

```csharp
public partial class login1 : System.Web.UI.Page
{
    lib obj = new lib();

    protected void Button1_Click(object sender, EventArgs e)
    {
```
//When the user clicks Log In. the username and password are checked and verified
against the table //'avi' for users and 'admin1' for administrator

```csharp
        if (TextBox1.Text == "Administrator")
        {
            obj.gettable("select pwd from admin1 where pwd = '" + TextBox2.Text + "' ");
            if (obj.dt.Rows.Count > 0)
            {
                Session["Name"] = "Administrator";
                Response.Redirect("welcome_admin.aspx");
            }
            else
            {
                Label4.Visible = true;
                Label4.Text = "Unauthorized Access";
            }
        }
        else
        {
            obj.gettable("select uname, pwd from avi where uname = '" + TextBox1.Text
+ "' and pwd = '" + TextBox2.Text + "'");

            if (obj.dt.Rows.Count > 0)
            {
```

```csharp
                Session["Name"] = obj.dt.Rows[0][0].ToString();
                Response.Redirect("welcome.aspx");
            }
            else
            {
                Label4.Visible = true;
                Label4.Text = "Invalid Username / Password";
            }
        }



    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        MailMessage mail = new MailMessage();
        obj.gettable("select email, pwd from avi where uname = '" + TextBox1.Text + "'
");
        mail.To.Add(obj.dt.Rows[0][0].ToString());
        mail.From = new MailAddress("aviiandevil@gmail.com");
        mail.Subject = "Password Recovery";
        mail.Body = obj.dt.Rows[0][1].ToString();
        mail.IsBodyHtml = true;

        SmtpClient smtp = new SmtpClient();
        smtp.Host = "smtp.gmail.com";
        smtp.Port = 587;
        smtp.UseDefaultCredentials = false;
        smtp.Credentials = new
System.Net.NetworkCredential("aviiandevil@gmail.com", "jonathandavis");
        smtp.EnableSsl = true;
        smtp.Send(mail);
        Label4.Visible = true;
        Label4.Text = "Password Sent!";
```

```
    }
    protected void admin_Click(object sender, EventArgs e)
    {
        TextBox1.Text = "Administrator";
    }

}
```

ISSUE BOOKS WORKING

```csharp
public partial class _Default : System.Web.UI.Page
{
    lib obj = new lib();

    protected void Page_Load(object sender, EventArgs e)
    {
        Label4.Text = Session["Name"].ToString();
        obj.gettable("select * from book");
        if (obj.dt.Rows.Count > 0)
        {
            GridView1.DataSource = obj.dt;
            GridView1.DataBind();
        }
    }


    protected void Button1_Click(object sender, EventArgs e)
    {


        TextBox2.Visible = true;
        obj.gettable("select * from account where uname = '" + Label4.Text + "' and bid
= " + TextBox1.Text + " ");

        if(obj.dt.Rows.Count > 0)
        {
            TextBox2.Text = "You cannot Issue a copy of the same book again";
        }
        else
        {
```

```
    obj.gettable("select bname, quantity from book where bid = "+TextBox1.Text+"
and status = 'Available' ");


     if (obj.dt.Rows.Count > 0)
    {
      TextBox2.Text = obj.dt.Rows[0][0].ToString() + " " + "ISSUED";
      TextBox3.Text = obj.dt.Rows[0][1].ToString() ;
      obj.excquery("insert into account values('" + Label4.Text + "', " +
TextBox1.Text + ", '" +System.DateTime.Now.ToLongDateString()+ "')");


      obj.excquery("update book set quantity = "+TextBox3.Text+" - 1  where bid = "
+ TextBox1.Text + " ");
      // obj.excquery("update book set quantity = " + Label3.Text + " + 1  where bid =
" + idbox.Text + " ");
        if(TextBox3.Text == "1")



        { obj.excquery("update book set status = 'Exhausted' where bid = " +
TextBox1.Text + " "); }



    }
    else
      TextBox2.Text = "NOT AVAILABLE";
    }



    obj.gettable("select * from book");
    if (obj.dt.Rows.Count > 0)
    {
      GridView1.DataSource = obj.dt;
      GridView1.DataBind();
    }
```

```
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Session.Abandon();
        Response.Redirect("login1.aspx");
    }
}
```

```
public partial class remove_books : System.Web.UI.Page
{
    lib obj = new lib();
    int dbse1, inp;
    protected void Page_Load(object sender, EventArgs e)
    {
        Label2.Text = Session["Name"].ToString();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        TextBox2.Visible = true;
        obj.gettable("select quantity from book where bid = " + TextBox1.Text + "");
        if (obj.dt.Rows.Count > 0)
        {
            dbse1 = int.Parse(obj.dt.Rows[0][0].ToString());
            inp = int.Parse(TextBox3.Text);
            if (inp > dbse1)
            {
                TextBox2.Text = "Quantity Error!!!";
            }
            else
            {
                TextBox2.Text = "Book Deleted";
                obj.excquery("update book set quantity = quantity - " + TextBox3.Text + "
where bid = " + TextBox1.Text + " ");
                obj.gettable("select quantity from book where bid = " + TextBox1.Text + "
");
                if (obj.dt.Rows.Count > 0)
                {
                    if (obj.dt.Rows[0][0].ToString() == "0")
```

```
                {
                    obj.excquery("delete from book where bid = " + TextBox1.Text + " ");
                    obj.excquery("delete from account where bid = " + TextBox1.Text + "
");
                }
            }
            GridView1.DataBind();
            GridView2.DataBind();
        }}


        else
        {
          TextBox2.Text = "Book Not Found";
        }
    }}
```

ACCOUNT WORKING

```csharp
public partial class _Default : System.Web.UI.Page
{
    lib obj = new lib();
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text=Session["Name"].ToString();
        obj.gettable("select ImageUrl from pic where uname = '" + Label1.Text + "'");
        if (obj.dt.Rows.Count > 0)
        {
            Image1.ImageUrl = obj.dt.Rows[0][0].ToString();
        }
        else
        {
            Image1.ImageUrl = "./up.png";
        }


    }


    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        obj.gettable("select ImageUrl from pic where uname = '" + Label1.Text + "'");
        if (obj.dt.Rows.Count > 0)
        {
            obj.excquery("delete from pic where uname = '" + Label1.Text + "'");
        }
```

```csharp
        if (FileUpload1.HasFile)
        {
            string fn = Path.GetFileName(FileUpload1.FileName);
            FileUpload1.SaveAs(Server.MapPath("./" + fn));
            Image1.ImageUrl = FileUpload1.PostedFile.FileName;
            obj.excquery("insert into pic values('" + Label1.Text + "', '" +
Image1.ImageUrl + "')");
        }
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        obj.gettable("select pwd from avi where uname = '" + Session["Name"] + "' and
pwd = '" + oldpwd.Text + "' ");
        if (obj.dt.Rows.Count > 0)
        {

            obj.excquery("update avi set pwd = '" + chpwd.Text + "' where uname = '" +
Session["Name"] + "' ");
            TextBox1.Visible = true;
            TextBox1.Text = "Password Updated!";
        }
        else
        {
            TextBox1.Visible = true;
            TextBox1.Text = "Old Password Invalid!";
        }

    }
}
```

```csharp
protected void Button1_Click(object sender, EventArgs e)
    {
      obj.gettable("select bid from account where uname = '" + TextBox1.Text + "' ");
      if (obj.dt.Rows.Count > 0)
      {
        for (i = 0; i < obj.dt.Rows.Count; i++)
        {
          obj.excquery("update book set quantity = quantity + 1 where bid = " +
obj.dt.Rows[i][0].ToString() + " ");
          obj.excquery("delete from account where uname = '" + TextBox1.Text + "'
");
        }
      }
      obj.excquery("delete from avi where uname = '" + TextBox1.Text + "' ");


      GridView1.DataBind();
      GridView2.DataBind();
      GridView3.DataBind();


    }
}
```

ADD BOOKS

```csharp
protected void Button1_Click(object sender, EventArgs e)
{

    obj.gettable("select quantity from book where bid = " + bookid.Text + " ");
    if (obj.dt.Rows.Count > 0)
    {
        if (obj.dt.Rows[0][0].ToString() == "0")
        {
            obj.excquery("update book set status = 'Available' where bid = " +
bookid.Text + " ");
        }


        obj.excquery("update book set quantity = " + obj.dt.Rows[0][0].ToString() + "
+ " + quantity.Text + " where bid = " + bookid.Text + " ");


    }
    else
    {
        Response.Write("No Such Book Record Exists. Please try Option 2");
    }

    quantity.Text = " ";
    GridView1.DataBind();


}
protected void Button2_Click(object sender, EventArgs e)
{
    obj.gettable("select * from book where bid = "+bookid2.Text+" ");
    if (obj.dt.Rows.Count > 0)
    {
```

```csharp
        Response.Write("Try option 1 or another BOOKID. A book with same
BookID already exists!");
    }
    else
    {
        obj.excquery("insert into book values(" + bookid2.Text + ", '" +
bookname.Text + "', 'Available', " + quantity2.Text + ")");
    }
    bookid2.Text = " ";
    bookname.Text = " ";
    quantity2.Text = " ";
    GridView1.DataBind();
  }
}
```

REMOVE BOOKS

```
public partial class remove_books : System.Web.UI.Page
{
    lib obj = new lib();
    int dbse1, inp;
    protected void Page_Load(object sender, EventArgs e)
    {
        Label2.Text = Session["Name"].ToString();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        TextBox2.Visible = true;
        obj.gettable("select quantity from book where bid = " + TextBox1.Text + "");
        if (obj.dt.Rows.Count > 0)
        {
            dbse1 = int.Parse(obj.dt.Rows[0][0].ToString());
            inp = int.Parse(TextBox3.Text);
            if (inp > dbse1)
            {
                TextBox2.Text = "Quantity Error!!!";
            }
            else
            {
                TextBox2.Text = "Book Deleted";
                obj.excquery("update book set quantity = quantity - " + TextBox3.Text + "
where bid = " + TextBox1.Text + " ");
                obj.gettable("select quantity from book where bid = " + TextBox1.Text + "
");
                if (obj.dt.Rows.Count > 0)
                {
                    if (obj.dt.Rows[0][0].ToString() == "0")
                    {
```

```csharp
                    obj.excquery("delete from book where bid = " + TextBox1.Text + " ");
                    obj.excquery("delete from account where bid = " + TextBox1.Text + "
");
                }
            }



        GridView1.DataBind();
        GridView2.DataBind();


    }
}


    else
    {
        TextBox2.Text = "Book Not Found";
    }

  }
}
```
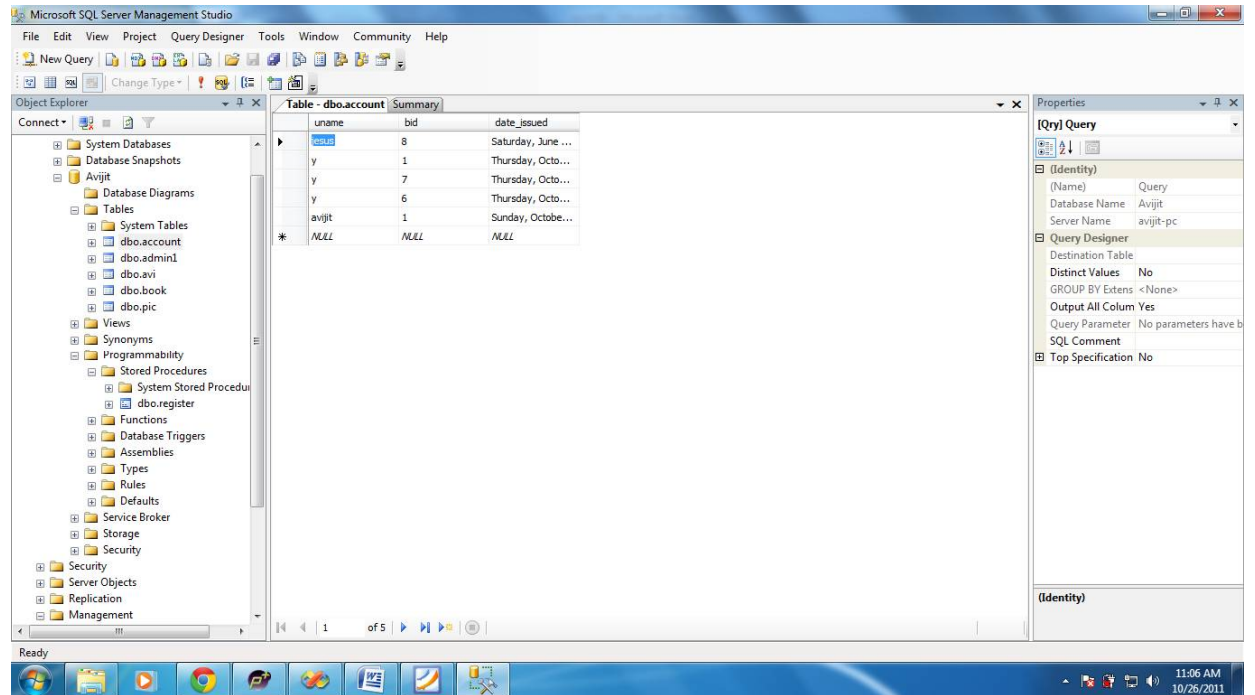
SERVER MANAGEMENT STUDIO

Tables and stored procedure created in the database

Stored Procedure and Tables

Register

USE [Avijit]

GO

/****** Object: StoredProcedure [dbo].[register]   Script Date: 03/30/2012 08:39:30

******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

ALTER procedure [dbo].[register](@uname varchar(50), @pwd varchar(50), @email

varchar(50), @question varchar(50), @answer varchar(50))

as

insert into avi values(@uname, @pwd, @email, @question, @answer)

Account Table

It holds the details of book issued by a person

Its functioning is contributed in account page of the website

CREATE TABLE [dbo].[account](

     [uname] [varchar](50) NULL,

     [bid] [int] NULL,

     [date_issued] [varchar](50) NULL

)

Admin1 Table

It is responsible for taking care of the administrator login details

CREATE TABLE [dbo].[admin1](

       [uname] [varchar](50) NULL,

       [pwd] [varchar](50) NULL

)

Avi Table

It holds all the details of the registered members

CREATE TABLE [dbo].[avi](

       [uname] [varchar](50) NULL,

       [pwd] [varchar](50) NULL,

       [email] [varchar](50) NULL,

       [question] [varchar](50) NULL,

       [answer] [varchar](50) NULL

)

Book Table

Responsible for holding the status of each and every book
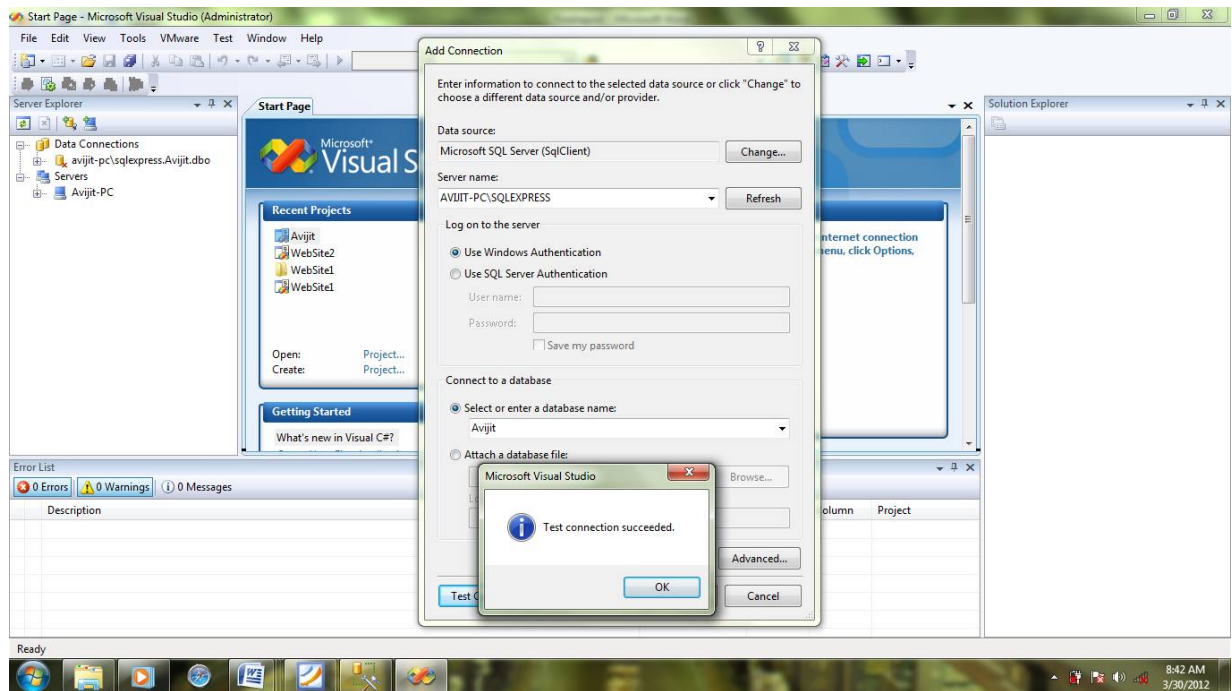
Used in almost whole of website

CREATE TABLE [dbo].[book](

       [bid] [int] NOT NULL,

       [bname] [varchar](50) NULL,

       [status] [varchar](50) NULL,

       [quantity] [int] NULL

)

Pic Table

Holds the location of the profile pictures of the users and the admin

```
CREATE TABLE [dbo].[pic](
        [uname] [varchar](50) NULL,
        [ImageUrl] [varchar](50) NULL
)
```

Connecting to the database

1. Right clicking 'Add connection on the data connections tab in server explorer
2. Selecting Microsoft sql server as the data source
3. Selecting the correct server name and database name

CONCLUSION

After we completed the project we were sure the problems in the existing system are overcome. The **"LIBRARY MANAGEMENT SYSTEM"** process made computerized to reduce human errors and to increase the efficiency. The main focus of this project is to lessen human efforts. The maintenance of the records is made efficient, as all the records are stored in the ACCESS database, through which data can be retrieved easily. The editing is also made simpler. The user has to just type in the required field and update the desired field.

Our main aim of the project is to get the correct information about a particular student and books available in the library.

The problems, which existed, have been removed to a large extent. And it is expected that this project will go a long way in satisfying user's requirements. The computerization of the Library Management will not only improve the efficiency but will also reduce human stress thereby indirectly improving human recourses

REFERENCES

- o Notes of Ms. Biji Elizabeth
- o Website for usage of templates: www.freewebtemplates.com
- o For querying a doubt: www.google.com
- o For knowledge on ASP.net through www.roseindia.net